韓國 컴퓨터情報學會 論文誌
第11卷 第5號, 2006. 11.

2006-11-5-1-3

XML 뷰 트리 기반의 XML 질의 처리 모델

정 채 영*, 김 현 주**

An XML Query Processing Model based on XML View Tree

Chai-Young Jung*, Hyun-Ju Kim**

요 약

본 논문에서는 XML 뷰 트리 기반 랩퍼의 질의 처리 모델을 제시한다. 질의 처리과정은 뷰 합성, 지역 정보원에 대한 질의 변환, 그리고 결과 문서 생성으로 이루어지며, 이를 위해 XML 뷰와 XML 질의어를 XML 뷰 트리로 표현하는 XML 뷰 트리 기반의 질의 처리 모델을 제시한다. XML 뷰 트리는 가상의 XML 문서의 구조를 그대로 반영하기 때문에 경로식의 탐색이 쉽고, XML 뷰에 대응되는 XML 스키마 생성과 질의 결과 문서 생성을 위한 템플릿으로 사용될 수 있다. 또한, XML 뷰와 XML 질의의 XML 뷰 트리를 통한 개념적 통일은 다단계 XML 뷰 정의와 합성을 쉽게 지원한다.

Abstract

This paper presents a query processing model in a wrapper based on the XML view tree. The query processing in a wrapper requires view composition, query translation into local sources, and generation of XML documents from local query results. We present a query processing model based on the view tree, where the XML views and the XML query is represented by the view tree. Since the view tree keeps the structure of a virtual XML document, it is easy to navigate the path expression. The view tree is also used as a template for schema generation and XML document generation as a query result. Moreover this conceptual uniform abstraction for the XML view and the user query makes it easy to support a multi-level XML view and to implement our composition mechanism.

▸ Keyword : XML, XML 뷰(XML View), XML 뷰 트리(XML View Tree), 사용자 정의 XML 뷰 (User-defined XML View), XML 질의처리(XML Query Processing), 데이터베이스 (Database), 랩퍼(Wrapper)

• 제1저자 : 정채영 • 교신저자 : 김현주
• 접수일 : 2006.10.20, 심사일 : 2006.11.05, 심사완료일 : 2006.11.12
* 진주산업대학교 경남IT엘리트사업단 PM ** 진주산업대학교 컴퓨터공학부 조교수

Ⅰ. 서 론

다양한 이질의 분산된 정보원을 모두 XML 정보원으로 보는 관점을 제공하면 이질의 정보원 모델에 대해 독립적인 질의가 가능하다. 이질의 정보원들을 XML 정보원처럼 사용하기 위해서는 각 정보원을 XML 관점으로 변환하는 XML 기반의 랩퍼가 필요하다. 랩퍼는 정보원의 스키마를 XML 뷰로 변환하여 사용자에게 제공하고, 사용자는 모든 정보원을 가상적인 XML 정보원으로 간주하여 XML 질의어를 이용하여 질의하면, 랩퍼는 질의 결과로 XML 문서를 전달한다[1, 2].

질의를 처리하기 위해서는 XML 뷰 기반의 질의를 각 정보원에 대한 스키마를 직접적으로 XML 스키마로 사상한 결과인 기본 XML 뷰에 대한 질의로 변환되어야 한다. 이 과정을 질의어 합성이라 한다. 합성된 질의는 지역 정보원의 질의로 변환한다. 그리고 그 결과는 XML 문서로 변환되어 반환된다. 따라서 질의 처리 과정은 질의어 합성, 변환, 결과 문서 생성의 세 가지 과정을 거친다[3, 4].

본 논문에서는 이 모든 과정을 하나의 모델로 처리할 수 있는 XML 뷰 트리 모델을 제시한다. 즉, 정보원의 스키마를 가상의 XML 스키마로 표현한 기본 XML 뷰, 기본 XML 뷰를 기반으로 정의되는 사용자 정의 XML 뷰, 그리고 가상의 XML 정보원에 대한 질의어 모두를 개념적으로 동일시하여, 그 표현은 가상적 혹은 실제적 XQuery로써, 그리고 내부 모델은 XML 뷰 트리로 정의한다. 그리고 질의 처리는 XML 뷰의 경로식에 대한 탐색를 통해서 이루어진다. XML 뷰 트리는 임의의 XQuery 표현식으로부터 구성되며, 가상의 XML 문서의 구조를 그대로 반영한다. 제시된 모델은 XML 뷰에 대한 경로식의 탐색을 통하여 합성 과정을 쉽게 수행하고, XML 뷰 트리를 질의 결과로 생성되는 XML 문서를 구성하기 위한 템플릿으로 활용할 수 있게 하며, XML 뷰에 대응되는 스키마 생성을 쉽게 한다. 개념적으로 추상화된 XML 뷰와 질의는 동일한 내부 모델로 표현되기 때문에 구현의 중복이 제거되며, 다단계로 정의되는 XML 뷰를 쉽게 지원한다.

본 논문에서는 XML 뷰와 질의를 표현하는 언어로서 W3C에서 제안한 XQuery[5, 6, 7]를 사용한다. 사용자가 XML 뷰를 정의하거나 질의를 하기 위해서는 XML 뷰에 대응되는 스키마가 필요하며, 랩퍼는 XML 뷰에 대한 스키마를 자동으로 생성하여 사용자에게 제공한다. XML 스키마를 표현하기 위한 언어로는 W3C 표준안인 XML Schema[8, 9, 10]를 사용한다. 그리고 제안한 질의처리 모델을 관계형 데이터베이스에 대하여 구현하고 그 결과를 제시한다. 그러나 제시된 질의처리 모델은 XML 뷰에 대해 개념적으로 추상화되어 있어 정보원 모델에 독립적이다. 요약하면, 첫째, 스키마 생성, 뷰 합성, 질의 변환, 결과 문서 생성을 모두 쉽게 할 수 있는 개념적으로 추상화된 트리 기반의 내부 데이터 모델을 제시하고, 둘째, 이 모델을 기반으로 하는 질의 처리 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 XML 뷰와 뷰 트리 모델에 대해 제시한다. 4장에서는 뷰 트리 합성과 이를 이용한 질의 변환과 결과 문서 생성 방법을 논한다. 5장에서는 제시된 모델에 대한 성능평가를 보인고, 마지막으로 6장에서 결론을 보인다.

Ⅱ. 관련 연구

SilkRoute[1, 3]와 XPERANTO[2, 4, 11]는 관계형 데이터베이스의 내용을 XML 문서로 출판하는 미들웨어 시스템이다. 이들은 사용자가 XML 뷰를 이용하여 정의하는 가상의 XML 문서를 기반으로 동작한다. 이때 가상의 XML 문서는 실체화되지 않으며 가상의 XML 문서에 대한 사용자 질의는 실제 관계형 데이터베이스에서 사용하는 질의어인 SQL로 변환하는 과정이 필요하다. 관계형 데이터베이스를 기반으로 하는 이들 시스템들은 질의 변환 방법이 관계형 데이터베이스에 종속적이다.

SilkRoute에서는 XML 뷰를 정의하기 위해 자체적으로 정의한 선언적 질의어인 RXL(Relational to XML translation Language)를 제공한다. RXL은 SQL 부분과 XML-QL[12]의 조합으로 이루어진 질의어로 블록 구조, 중첩 질의, 그리고 스콜렘 함수 등을 지원한다. 사용자는 RXL을 이용하여 XML 뷰를 기술하고, XML-QL를 이용하여 질의를 한다. 가상의 XML 문서에 대한 문서구조를 유지하는 템플릿과 문서의 각 노드의 생성 규칙을 가지는 스콜렘 함수로 뷰 트리를 구성하며 템플릿에 대한 패턴 매칭과 스콜렘 함수에 대한 질의 재작성을 통해 합성 과정을 수행한다. 합성을 통해 새로운 RXL 질의를 생성하며 이를 실행 가능한 질의(executable query)라 한다. 합성 결과로부터 중복되는 프레디키트와 조인들로 여러 가능성 있는 다수의 질의 플랜을 생성한다. 이때 최적의 질의 플랜을 생성하는 최적화 알고리즘은 NP-Complete이기 때문에 greedy 알고리즘을 사용한다. 선택한 질의 플랜에 의해 생성된 SQL을 수행하여

생성된 결과는 템플릿을 이용하여 통합한다.

XPERANTO는 XML 뷰와 사용자 질의를 XQuery를 사용하여 표현하며, 관계형 모델에서 사용하는 QGM을 확장한 XQGM(XML Query Graph Model)을 내부 질의 모델로 이용한다. XQuery로 기술된 XML 뷰와 사용자 질의는 XQGM으로 변환되어 합성되며 합성 후 새로운 XQGM을 생성한다. 합성 과정에서 사용자 질의의 프레디키트를 XML 뷰로 내려보낸 후, 사용자 질의에 나타난 XML 뷰에 대한 탐색을 제거하고, 사용자 질의에 포함되지 않은 부분은 XML 뷰에서 제거한다. 합성된 XQGM은 SQL로 변환되는 부분과 결과 XML 문서를 생성하는 부분으로 분리되어 각각 질의 변환과 결과 문서 생성에 이용된다. Sorted Outer Union 방법을 이용하여 하나의 최소화된 SQL을 생성하며 이를 수행한 결과를 이용하여 결과 문서를 생성한다.

본 논문은 XQuery를 이용하여 XML 뷰와 사용자 질의를 정의하고, XML 뷰와 사용자 질의를 추상화하여 가상적인 XML 문서의 구조를 그대로 반영하는 트리를 구성한다. 합성은 사용자 질의에 나타나는 경로식을 XML 뷰 트리에서 탐색하여 바인딩된 노드에 대한 포인터 연결을 유지하는 것으로 이루어진다. 본 논문에서는 SilkRoute와 달리 기본 XML 뷰를 지원하며 XML 뷰 정의어와 사용자 질의어가 동일하다. 또한 뷰 합성을 통해 하나의 최소화된 질의 플랜을 만들어 낸다. 그리고, XPERANTO와 달리 합성 과정에서 경로식에 대한 추적을 통해 XML 뷰 트리의 질의 정보를 사용자 질의 트리로 모은다. 이는 사용자 질의에 포함되지 않은 부분을 XML 뷰 트리에서 제거하는 과정이 필요하지 않고 한번 생성된 XML 뷰 트리를 다른 사용자 질의와의 합성에 그대로 이용할 수 있다. 또한 이런 연산은 기반 관계형 데이터베이스와 상관없이 이루어진다. 게다가, 제안하는 XML 뷰 트리는 합성뿐만 아니라 질의 변환, 결과 문서 생성에 그대로 사용되는 기본 프레임워크 역할을 한다.

III. XML 뷰와 XML 뷰 트리

3.1 기본 XML 뷰 트리의 생성

랩퍼는 지역 정보원에 저장되어 있는 전체 데이터들을 XML 문서 관점으로 표현한 XML 뷰를 자동적으로 생성하는데, 이것을 기본 XML 뷰라고 한다[4]. 본 논문에서는 [4]에서 제시한 것과는 달리 각각의 테이블을 각각의 기본 XML 뷰로 표현한다. 실체화되지 않는 기본 XML 뷰 인스턴스를 구성할 수 있는 가상적인 질의를 설정하고 이것을 바탕으로 기본 XML 뷰 트리를 자동적으로 구성한다. 기본 XML 뷰 트리의 생성에는 데이터베이스의 시스템 카탈로그가 이용되며, 질의 변환 및 스키마 생성에 필요한 시스템 카탈로그의 정보를 뷰 트리에 유지한다.

XML 뷰 트리는 XQuery의 엘리먼트 생성자로부터 만들어지는 엘리먼트의 정보를 저장하는 엘리먼트 노드와 FLWR(For-Let-Where-Return) 표현식의 질의 정보를 저장하는 FLWR 노드로 구성된다. 엘리먼트 노드는 엘리먼트의 이름과 데이터 타입, 엘리먼트의 텍스트 데이터와 연결되는 경로식 등이 저장된다. FLWR 노드에는 FLWR 절의 변수와 프레디키트, 정렬 조건에 대한 정보가 저장된다. FLWR 노드 하위에는 FLWR 절에 의해서 구성되는 엘리먼트 노드가 추가된다.

3.2 사용자 정의 XML 뷰와 사용자 질의

기본 XML 뷰를 기반으로 사용자가 정의하는 형태의 가상의 XML 문서로 구성할 수 있는데 이를 사용자 정의 XML 뷰라고 한다. 사용자 정의 XML 뷰의 개념은 [1, 2]에서 제시되었는데, 여기서는 그 개념을 그대로 활용한다. 본 논문에서 사용자 정의 XML 뷰는 XQuery를 이용하여 정의한다. 그림 1은 XQuery를 이용해 정의된 사용자 정의 XML 뷰의 예이다. 그림 2는 그림 1의 사용자 정의 XML 뷰의 이름이 Auction일 때, 문자열 "Kang"이 포함된 이름을 가지는 사람들의 이름과 입찰 리스트를 반환하는 질의이다. 사용자 정의 XML 뷰와 사용자 질의는 XQuery로 정의되기 때문에 사용자 정의 XML 뷰 트리와 질의 트리는 XQuery로부터 구성된다.

IV. XML 뷰의 합성과 질의 처리

4.1 XML 뷰 합성

그림 3은 기본 XML 뷰 트리에 그림 1에서 정의된 사용자 정의 XML 뷰와 그림 2의 사용자 질의가 합성된 그림이다. 그림 3의 아랫부분은 기본 XML 뷰 트리이고, 중간부분은 사용자 정의 XML 뷰 트리, 윗부분은 사용자 질의 트리이다.

사용자 질의와 XML 뷰의 합성을 위하여 사용자 정의 XML 뷰 트리를 구성할 때 그림 3에서처럼 사용자 정의

```
〈Auction〉 {
  for $user in document("users")/tuple
  where $user/rating = "A"
  return
    〈User ID={$user/id/text()}〉
      〈Name〉 $user/name/text() 〈/Name〉
        〈Bids〉 {
          for $item in document("items")/tuple, $bid in document("bids")/tuple
          where $bid/u_id = $user/id AND $bid/i_id = $item/id
          return
            〈Item〉
              〈Description〉 $item/description/text() 〈/Description〉
              〈Price〉 $item/price/text() 〈/Price〉
              〈Bid〉 $bid/bid/text() 〈/Bid〉
            〈/Item〉
        }
        〈/Bids〉
        〈Rating〉 $user/rating/text() 〈/Rating〉
    〈/User〉
}
〈/Auction〉
```

그림 1. 사용자 정의 XML 뷰
Fig 1. User-defined XML View

XML 뷰와 기본 XML 뷰 사이의 관계를 미리 구성한다. 이를 위하여 사용자 정의 XML 뷰 트리에 노드가 추가될 때, 그 노드에 연관된 경로식을 탐색하여 대응되는 노드에 대한 포인터가 유지된다. 그림 3에서 연결된 선이 이를 나타낸다.

```
for $u in document("Auction")/User
where contains($u/Name, "Kang")
return
〈Result〉
〈Name〉 $u/Name/text() 〈/Name〉
{ $u/Bids }
〈/Result〉
```

그림 2. 사용자 질의
Fig 2. User Query

XQuery에 나타나는 경로식에 대한 탐색은 실체화된 XML 문서에 대해 이루어지는 것이지만, XML 뷰 트리는 가상적인 XML 문서의 메타데이터이기 때문에 스키마 수준에서 탐색이 이루어진다. 결국, 뷰 트리에 대한 탐색의 결과는 실체화된 XML 문서에 포함된 엘리먼트의 메타데이터가 된다.

사용자 질의에 나타난 경로식 역시 사용자 정의 XML 뷰 트리에 대한 포인터로 유지된다. 그림 2의 질의에서 6번째 행의 숨은 경로식은 해당 엘리먼트와 그 하위 엘리먼트를 모두 질의 결과에 포함시키는 것이므로 경로식에 바인딩된 노드와 그 하위 노드를 복사하여 질의 트리에 추가한다. 그림 3에서 사용자 질의의 변수 $u가 사용자 정의 XML 뷰 트리의 User 엘리먼트 노드에 바인딩 되고 이 엘리먼트 노드는 사용자 정의 XML 뷰의 $user로부터 파생되므로 $u는 $user의 프레디키트에 영향을 받는다. 이것을 질의 트리에 반영하기 위하여 $user의 프레디키트를 질의 트리로 복사하여 $u의 프레디키트에 추가한다. 즉, 하위 뷰의 프레디키트를 상위의 질의로 끌어올린 것이다.

본 논문에서는 사용된 합성 메커니즘은 합성의 결과로 모든 질의 정보가 최상위의 질의 트리에 모이게 된다. 반대로 상위 질의의 프레디키트를 하위 뷰로 내려보내어 질의 정보를 하위의 뷰로 모으는 방법도 가능한데, 이 경우 최종 질의에 포함되지 않는 뷰의 요소를 합성된 뷰 트리로부터 제거해야 하며, 뷰 트리를 다른 질의와 공유할 수 없는 단점이 있다. 본 논문의 방법은 질의에 의해 구성되는 엘리먼트 노드들만 트리에 추가되며, 질의에 관련된 뷰의 프레디키트만 끌어올려지기 때문에 질의에 필요하지 않은 뷰의 요소는 처음부터 포함되지 않는다. 그리고 하위의 뷰 트리는 다른 상위의 질의들과 공유될 수 있다. 합성된 질의 트리는 질의 변환, 결과 문서 생성에 이용된다.

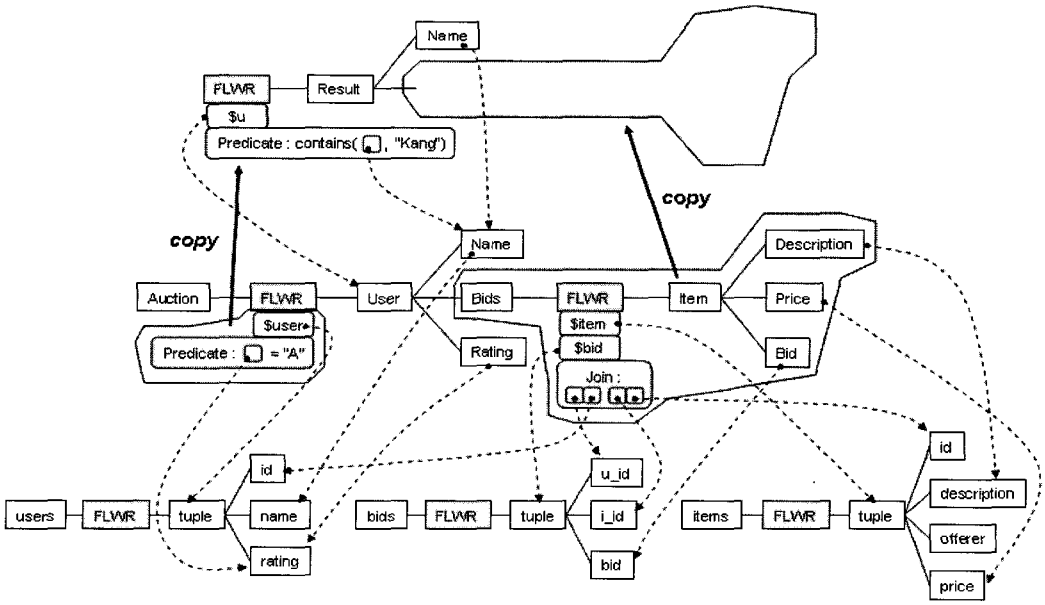

그림 3. XML 뷰 합성
Fig 3. Composition XML View

4.2 질의 변환

합성된 질의는 지역 정보원에서 수행 가능한 질의로 변환되어야 하는데, 여기서는 관계형 데이터베이스에 대해 질의어인 SQL로 변환하는 과정을 설명한다.

먼저 합성된 질의 트리를 순회하여 변환에 필요한 정보들을 FLWR 노드에 모은다. 이렇게 모여진 정보를 이용하여 FLWR 노드당 하나씩의 SQL 문장을 생성한다. SQL의 SELECT 절과 FROM 절을 구성하는 정보는 FLWR 노드의 하위 엘리먼트 노드에 바인딩된 기본 XML 뷰 트리의 노드로부터 얻고, 관련된 프레디키트는 SQL의 WHERE 절을 구성한다. 그림 4는 합성된 질의 트리를 기반으로 생성된 SQL 질의를 보여준다. 그림에서 질의 트리와 기본 XML 뷰 트리 사이의 사용자 정의 XML 뷰 트리는 생략되었다.

그림 4에서 상위의 FLWR 노드와 하위의 FLWR 노드는 서로 조인관계이다. 이렇게 조인된 질의를 처리하는 가장 직관적인 방법은 그림 4에서 번역된 SQL과 같이 호스트 변수를 사용하는 것이다. 이 방법은 상위의 SQL을 수행시키고 그 결과 튜플을 하나씩 하위 SQL의 호스트 변수에 적용시킴으로써 조인 연산을 수행한다. 결국, 조인 연산이 랩퍼에서 수행되며, 하위의 SQL이 여러 번 수행되므로 성능 저하를 초래 할 수 있다. 물론 2개의 SQL을 하나의 SQL로 합쳐서 수행시킬 수도 있는데, 이때는 중복되는 데이터로 인해 결과 문서를 생성하는데 어려움이 있다.

이러한 성능 저하를 해소하기 위해서는 가능하면 지역 정보원의 질의 처리 능력을 최대한 활용하는 것이 중요하다. 관계형 데이터베이스에서 조인으로 상호 연관된 질의를 디코릴레이션 시켜서 독립적으로 수행하게 하는 것은 널리 알려진 방법이다. 본 논문에서는 이를 상위의 FLWR 노드의 프레디키트를 하위의 FLWR 노드로 내려보내고 호스트 변수를 조인으로 대체함으로써 이루어진다. 이 때, 결과 XML 문서를 수월하게 구성할 수 있도록 질의를 변환하는 것 또한 중요한데, 이를 위해서 연관된 SQL 질의를 같은 조건으로 정렬한다.

그림 5은 디코릴레이션된 질의 트리와 이를 기반으로 생성된 SQL이다. 그림에서 상위 FLWR 노드의 프레디키트가 하위 FLWR 노드로 내려보내진 것을 확인할 수 있다. 그리고 하위 FLWR 노드에 생성된 SQL에 이 프레디키트가 반영되어 있고, 두 개의 SQL은 같은 정렬 조건을 가지고 있다. 두 개의 SQL은 각각 독립적으로 수행되고, 결과 튜플이 같은 조건으로 정렬되어진다.

4.3 결과 문서 생성

그림 6는 그림 5에서 생성된 질의를 수행한 결과와 이를

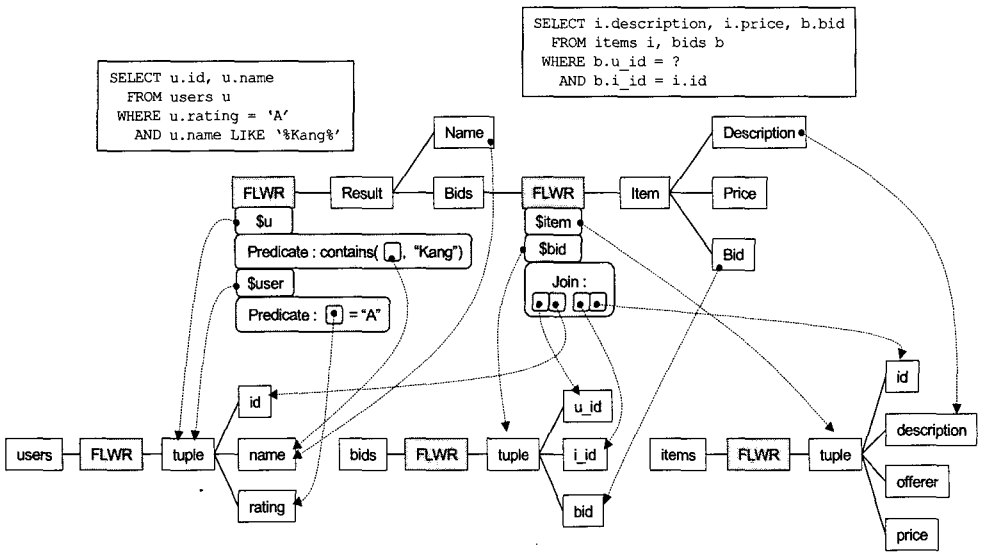

그림 4. 호스트 변수를 이용해 생성된 SQL
Fig 4. The Created SQL by Host Variable

이용해 생성된 결과 XML 문서를 나타낸다. 질의 트리가 결과 XML 문서의 템플릿 역할을 하기 때문에 결과 XML 문서를 생성하는 것은 직관적이다.

그림 6에서 최상위의 노드가 FLWR 노드이므로 번역된 SQL을 실행시키고, 결과 집합이 하나의 튜플을 가지고 있으므로 하위 노드를 한번 순회하게 된다. 따라서, 하위의 Result와 Name, 그리고 Bids 엘리먼트가 차례대로 결과 XML 문서에 추가되게 되는데, Name 엘리먼트의 경우에는 FLWR 노드의 결과 튜플이 적용된다.

Bids 엘리먼트 노드의 하위 FLWR 노드는 상위의 FLWR 노드와 조인되어지므로 결과 집합에 속한 튜플 중 현재 적용중인 상위 FLWR 노드의 결과 튜플과 연관된 튜플들만 하위의 노드에 적용된다. 결국, 두 연관된 결과 집합 사이의 외부 조인이 이루어지는 것인데, 각각의 결과 레코

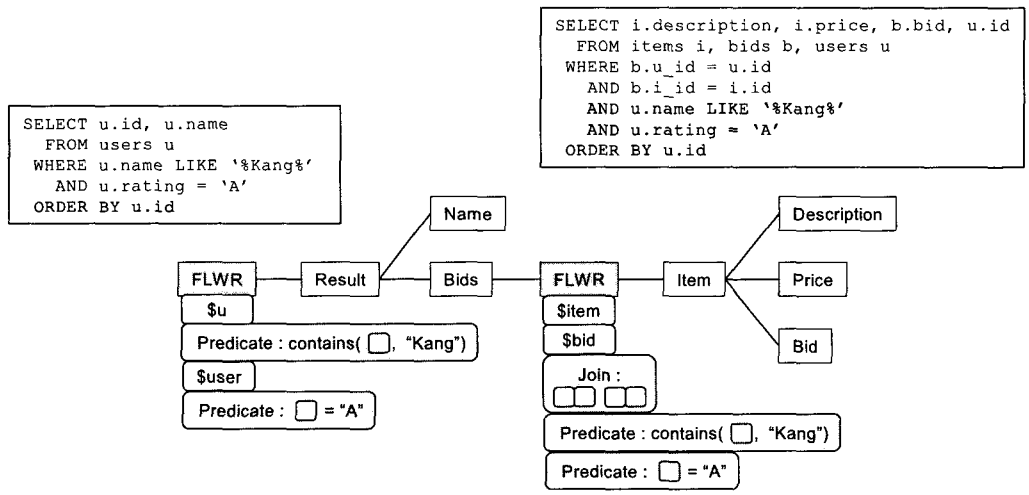

그림 5. 디코릴레이션을 이용해 생성된 SQL
Fig 5. The Created SQL by Decorrelation

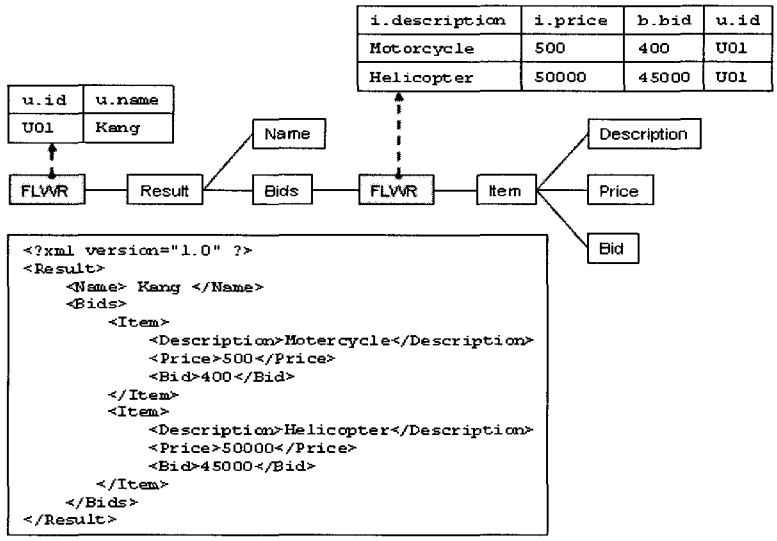

그림 6. SQL 수행 결과와 생성된 XML 문서
Fig 6. The Execution Result of SQL and Created XML Document

드 셋이 정렬되어져 있기 때문에 싱글 패스로 결과 XML 문서를 생성할 수 있다. 그림 6에서 하위 FLWR 노드의 결과 집합에 속한 두 개의 튜플이 모두 현재 적용중인 상위 FLWR 노드의 결과 튜플과 연관되므로, Item 엘리먼트 노드 이하는 두 번 순회하게 된다.

이와 같은 질의 변환 및 결과 문서 생성 메커니즘은 결과적으로 XPERANTO 연구에서 사용된 Sorted Outer Union(SOU)[11] 방법과 유사하다. SOU는 연관된 질의를 디코릴레이션하여 정렬하고 이것을 외부 조인으로 합치는 하나의 SQL 질의를 관계형 데이터베이스 엔진에서 수행하게 하는 방법인데 결과 튜플의 길이가 길어지는 단점이 있다. 본 논문의 방법은 랩퍼에서 생성한 디코릴레이션된 질의가 각각 독립적으로 관계형 데이터베이스에서 수행되고 태거에 의해서 외부 조인이 이루어지며, 각각의 결과 튜플에는 관련된 컬럼들만 포함되게 된다.

V. 실험 및 평가

제시된 모델은 관계형 데이터베이스에서 구현되었다. 구현 언어로 자바(JDK 1.4)를 사용하였으며, 대상 데이터베이스로 MS-SQL Server 2000을 사용하였다. 실험은 펜티엄3 1GHz CPU와 512M 메모리에서 작동하는 윈도우 2000 서버에서 수행되었다. 예측할 수 없는 네트워크의 딜레이를 피하기 위해 데이터베이스 역시 동일한 시스템에서 작동시켰다.

테스트를 위해 3개의 테이블을 가지는 데이터베이스에 대해 적은 양의 결과를 반환하는 질의와 상대적으로 많은 양의 결과를 반환하는 서로 다른 질의를 실행시켰다. 각 질의에는 조인으로 연관되는 내포된 FLWR 절을 가지고 있다. 또한, 질의 변환에 있어서 호스트 변수를 이용하는 방법과 디코릴레이션을 이용하는 방법을 모두 테스트하였다. 표 1은 두 질의를 여러 번 수행하여 걸린 평균 시간을 제시한 것인데, 질의를 파싱하고 합성하는데 걸린 시간, SQL로 변환하는데 걸린 시간, 그리고 변환된 질의를 수행하여 결과 문서를 생성하는데 걸린 시간을 측정한 결과이다. 시간의 단위는 1/1000초(mili-second)이다.

표 1에서 질의를 파싱하고 합성하고 SQL로 번역하는 시간은 양쪽 질의 모두 약 0.35초 정도로서 무시할 수 있는 시간이라고 할 수 있다. 그러나 결과 문서를 생성하는 부분에서 결과 문서의 크기가 수행 시간에 영향을 미치는 것을 알 수 있다. 결과 문서를 생성하는 데 있어서 호스트 변수를 사용하는 경우와 디코릴레이션을 사용하는 경우를 비교하면 디코릴레이션을 사용하는 방법이 더 나은 성능을 보여주고 있으며, 두 번째 질의의 경우에 13만 라인의 3.5MB 정도의 결과 문서를 생성하는데 있어서 1초 정도의 만족할 만한 성능을 보여주고 있다. 그리고 예상대로 호스트 변수를 사용하는 경우는 결과 문서의 크기가 커질수록 수행 시간이 더 크게 증가한다.

표 1. 질의 수행 시간 및 결과 문서 크기 분석
Table 1. The Analysis of Query Execution Time and Result XML Document size

질의	Parsing and Composition Time	Translation Time	Tagging Time (Using Host Variable)	Result XML Document Size (Line Count)
Q1	337	10	290(350)	46KB (2,186)
Q2	348	10	1,065(3,265)	3,650KB (130,000)

VI. 결론

XML 뷰 기반의 랩퍼 시스템은 이질의 분산된 정보원들을 모두 XML 정보원으로 간주하여 사용자로 하여금 XML 정보원처럼 사용할 수 있도록 한다. 이질의 정보원을 XML 정보원처럼 사용하기 위해서는 지역 정보원의 스키마를 XML 뷰로 정의하는 것이 중요하다. 각 지역 정보원의 스키마를 XML 뷰로 변환하여 사용자에게 제공하면, 사용자는 이질의 정보원들을 모두 XML 정보원으로 간주하여 XML 질의어를 이용하여 질의를 하게 됨으로서 이질적인 정보원들의 모델에 독립적인 질의를 가능하게 한다.

본 논문에서는 XML 뷰 기반의 질의 처리 모델과 이를 기반으로 한 질의 처리 메커니즘을 제시하였다. 이를 위해서 기본 XML 뷰와 사용자 정의 XML 뷰, 그리고 사용자 질의어에 대한 개념적인 추상화를 통하여 모든 XML 뷰를 동일한 데이터 모델로 표현한 트리 기반의 질의 처리 모델을 제시하였다. 이를 기반으로 랩퍼에서는 뷰 합성, 지역 정보원에 대한 질의 변환, 그리고 결과 문서 생성 과정을 수행한다. 본 논문에서 제시한 XML 뷰 기반의 질의 처리 모델은 XML 뷰와 사용자 질의가 내부적으로 동일한 트리 모델을 이용하여 표현되기 때문에 구현의 중복을 피할 수 있으며, 뷰 트리를 이용한 경로 탐색이 용이하여 질의어 합성을 편하게 한다. 그리고, 다단계로 정의되는 사용자 정의 XML 뷰를 쉽게 지원할 수 있다.

본 논문에서 제시한 XML 뷰 기반의 질의 처리 모델은 지역 정보원에 독립적이기 때문에 다른 이질적인 지역 정보원에 대해서도 적용이 가능한 범용적인 모델이다. 따라서, 본 논문에서 제시한 트리 모델과 이를 기반으로 하는 합성 메커니즘은 XML 뷰를 지원하는 랩퍼를 구현하기 위한 프레임워크로 활용될 수 있으며, 미디에이터 기반의 이질의 정보원을 통합하기 위한 미들웨어 시스템을 개발하는데 활용 가능하다.

참고문헌

[1] M. Fernandez, W. Tan, and D. Suciu, "SilkRoute : Trading between Relations and XML," Proceedings of the 9th International World Wide Web Conference (WWW9), pp. 723-745, 2000.

[2] M. Carey, D. Florescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita, and S. Subramanian, "XPERANTO: Publishing Object-Relational Data as XML," Proceedings of the International Workshop on the Web and Databases (WebDB), pp. 105-110, May 2000.

[3] M. Frenandez, A. Morishima, and D. Suciu, "Efficient Evaluation of XML Middle-ware Queries," Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp. 103-114, 2001.

[4] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan, and J. Funderburk, "Querying XML Views of Relational Data," Proceedings of the 27th International Conference on Very Large Data Bases (VLDB '2001), pp. 261-270, 2001.

[5] World-Wide Web Consortium, "XQuery 1.0: An XML Query Language," http://www.w3.org/TR/xquery/, W3C Candidate Recommendation, 2006

[6] World-Wide Web Consortium, "XQuery 1.0 and XPath 2.0 Data Model," http://www.w3.org/TR/query-datamodel/, W3C Candidate Recommendation, 2006.

[7] World-Wide Web Consortium, "XML Query Use Cases," http://www.w3.org/TR/xquery-use-

cases/, W3C Working Draft, 2006.
[8] World-Wide Web Consortium, "XML Schema Part 0: Primer," http://www.w3.org/TR/xml-schema-0/, W3C Recommendation, 2004.
[9] World-Wide Web Consortium, "XML Schema Part 1: Structures," http://www.w3.org/TR/xmlschema-1/, W3C Recommendation, 2004.
[10] World-Wide Web Consortium, "XML Schema Part 2: Datatypes," http://www.w3.org/TR/xmlschema-2/, W3C Recommendation, 2004.
[11] J. Shanmugasundaram, E. Shekita, R. Barr, M. Carey, B. Lindsay, H. Pirahesh, and B. Reinwald, "Efficiently Publishing Relational Data as XML Documents," VLDB Journal, Vol 10, No 2-3, pp.133-154, 2001.
[12] World-Wide Web Consortium, "XML-QL: A Query Language for XML," http://www.w3.org/TR/NOTE-xml-ql/, 1998.

저 자 소 개

정 채 영

1997년 경상대학교 전자계산학과(이학사)
2001년 경상대학교 컴퓨터과학과(공학석사)
2004년 경상대학교 컴퓨터과학과(공학박사)
2004년 ~ 현재 진주산업대학교 컴퓨터공학과 시간강사
관심분야 : XML, 웹 프로그래밍, 데이터베이스, 정보통합

김 현 주

1988년 경상대학교 전자통계학과(이학사)
1990년 숭실대학교 전자계산학과(공학석사)
2000년 경상대학교 전자계산학과(공학박사)
1994년 ~ 1997년 제일정밀공업(주) 연구원
2000년 ~ 2002년 경남정보대학 컴퓨터정보계열 전임강사
2002년 ~ 현재 진주산업대학교 컴퓨터공학부 조교수
관심분야 : 정보검색, 디지털 도서관, 웹 프로그래밍, XML